Refine Search

Search Results -

Terms	Documents
L5 and @pd > 20050515	3

Database:

> US Pre-Grant Publication Full-Text Database
> US Patents Full-Text Database
> US OCR Full-Text Database
> EPO Abstracts Database
> JPO Abstracts Database
> Derwent World Patents Index
> IBM Technical Disclosure Bulletins

Search:

> L6

Refine Search

Recall Text   Clear   Interrupt

Search History

DATE: Sunday, June 19, 2005   Printable Copy   Create Case

Set Name Query	Hit Count	Set Name
side by side		result set

DB=USPT; PLUR=NO; OP=OR

Set Name	Query	Hit Count	Set Name
L6	L5 and @pd > 20050515	3	L6
L5	(717/130).ccls.	184	L5
L4	L2 AND XML	0	L4
L3	L2 AND (instrument OR profile Or monitor OR filter)	23	L3
L2	XAM	54	L2
L1	XAM ADJ architecture	0	L1

END OF SEARCH HISTORY

Hit List

Search Results - Record(s) 1 through 3 of 3 returned.

☐ 1. Document ID: US 6904594 B1

L6: Entry 1 of 3   File: USPT   Jun 7, 2005

US-PAT-NO: 6904594
DOCUMENT-IDENTIFIER: US 6904594 B1

TITLE: Method and system for apportioning changes in metric variables in an symmetric multiprocessor (SMP) environment

DATE-ISSUED: June 7, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Berry; Robert Francis	Austin	TX		
Howard; John Day	Austin	TX		
Levine; Frank Eliot	Austin	TX		
Urquhart; Robert J.	Austin	TX		

US-CL-CURRENT: 718/100; 717/130, 717/158, 718/102, 718/107

ABSTRACT:

A method and system for monitoring performance of a program using global metric variables to provide the support in an symmetric multiprocessor (SMP) system. A Java virtual machine (Jvm) either calls the profiler whenever bytes are allocated or provides an interface to allow the profiler to determine the value of the change in the metric for the current thread. The profiler then applies the changes to a metric for the current thread. Alternatively, per processor data areas are maintained for storing per processor metric values. Whenever a thread switch occurs or there is a request for the metric on a specified thread, an operating system kernel updates the thread level metric values with changes in the values per processor metrics.

43 Claims, 41 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

| Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw D |

☐ 2. Document ID: US 6898785 B2

L6: Entry 2 of 3   File: USPT   May 24, 2005

US-PAT-NO: 6898785
DOCUMENT-IDENTIFIER: US 6898785 B2

TITLE: Handling calls from relocated instrumented functions to functions that
expect a return pointer value in an original address space

DATE-ISSUED: May 24, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ramasamy; Vinodha	Campbell	CA		
Gouriou; Eric	Sunnyvale	CA		
Hundt; Robert	Sunnyvale	CA		

US-CL-CURRENT: <u>717</u>/<u>129</u>; <u>714</u>/<u>34</u>, <u>717</u>/<u>130</u>, <u>717</u>/<u>131</u>, <u>717</u>/<u>133</u>

ABSTRACT:

Method and apparatus for handling calls from relocated instrumented functions to
functions that expect a return pointer value in an original address space. In
various embodiments of the invention, instrumented versions of selected functions
of an executable program are generated and stored in a relocation address space.
When a function is called by a function in the relocation address space, a return
pointer register stores a first return-pointer value that is an address in the
relocation address space. The address in the original address space that
corresponds (logically) to the first return-pointer value is identified as an
original return-pointer value. The first return-pointer value is associated with
the original return-pointer value, references to the original return-pointer value
are substituted for references to the first return-pointer value, and the
instruction at the address indicated by the original return-pointer value is
replaced with a breakpoint. When the breakpoint is encountered upon return of
control at the original return-pointer value, the first return-pointer value that
is associated with the original return-pointer value is obtained, and control is
transferred to the instruction at the address referenced by the first return-
pointer value.

14 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D |

☐ 3. Document ID: US 6895578 B1

L6: Entry 3 of 3   File: USPT   May 17, 2005

US-PAT-NO: 6895578
DOCUMENT-IDENTIFIER: US 6895578 B1

TITLE: Modularizing a computer program for testing and debugging

DATE-ISSUED: May 17, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kolawa; Adam K.	Bradbury	CA		
Byers; Chad E.	San Luis Obispo	CA		

US-CL-CURRENT: 717/130; 717/134, 717/143

ABSTRACT:

A system and method for facilitating and simplifying testing and debugging of computer programs. is described A computer program is broken down to smaller components, such as, classes, functions, or objects, and then those smaller components are tested individually. Accordingly, specific aspects of the computer program can be effectively tested. The user can automatically perform a range of tests on a class or method when the class or method is compiled without integrating the class or method into a larger project.

37 Claims, 30 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 27

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D |

| Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS |

Terms	Documents
L5 and @pd > 20050515	3

Display Format: REV | Change Format |

Previous Page   Next Page   Go to Doc#